EXHIBIT B10

WIKIPEDIA
The Free Encyclopedia

Apache Subversion

Apache Subversion (often abbreviated **SVN**, after its command name *svn*) is a software versioning and revision control system distributed as open source under the Apache License.[2] Software developers use Subversion to maintain current and historical versions of files such as source code, web pages, and documentation. Its goal is to be a mostly compatible successor to the widely used Concurrent Versions System (CVS).

The open source community has used Subversion widely: for example, in projects such as Apache Software Foundation, Free Pascal, FreeBSD, SourceForge, and from 2006 to 2019, GCC. CodePlex was previously a common host for Subversion repositories.

Subversion was created by CollabNet Inc. in 2000, and is now a top-level Apache project being built and used by a global community of contributors.[3]

History

CollabNet founded the Subversion project in 2000 as an effort to write an open-source version-control system which operated much like CVS but which fixed the bugs and supplied some features missing in CVS.[4] By 2001, Subversion had advanced sufficiently to host its own source code,[4] and in February 2004, version 1.0 was released.[5] In November 2009, Subversion was accepted into Apache Incubator: this marked the beginning of the process to become a standard top-level Apache project.[6] It became a top-level Apache project on February 17, 2010.[7]

Apache Subversion

Original author(s)	CollabNet
Developer(s)	Apache Software Foundation
Initial release	20 October 2000
Stable release(s) [±] (https://en.wikipedia.org/wiki/Template:Latest_stable_software_release/Subversion?action=edit)	
1.14.x LTS	1.14.2 / 12 April 2022[1]
Preview release(s) [±] (https://en.wikipedia.org/wiki/Template:Latest_preview_software_release/Subversion?action=edit)	
Repository	svn.apache.org/repos/asf/subversion/trunk/ (https://svn.apache.org/repos/asf/subversion/trunk/)
Written in	C
Operating system	Centos · Debian · Fedora · FreeBSD · HP-UX · NetBSD · OpenBSD · openSUSE · OpenVMS ·

	macOS · Red Hat Linux · Solaris · SUSE Linux · Ubuntu · Windows
Type	Revision control
License	Apache-2.0[a]
Website	subversion.apache.org (https://subversion.apache.org/)

Version	Original release date	Latest version	Release date	Status
1.0	2004-02-23	1.0.9	2004-10-13	No longer supported
1.1	2004-09-29[8]	1.1.4	2005-04-01	No longer supported
1.2	2005-05-21[9]	1.2.3	2005-08-19	No longer supported
1.3	2005-12-30[10]	1.3.2	2006-05-23	No longer supported
1.4	2006-09-10[11]	1.4.6	2007-12-21	No longer supported
1.5	2008-06-19[12]	1.5.9	2010-12-06	No longer supported
1.6	2009-03-20[13]	1.6.23	2013-05-30	No longer supported
1.7	2011-10-11[14]	1.7.22	2015-08-12	No longer supported
1.8	2013-06-18[15]	1.8.19	2017-08-10	No longer supported
1.9	2015-08-05[16]	1.9.12	2019-07-24	No longer supported
1.10	2018-04-13[17]	1.10.8	2022-04-12	No longer supported
1.11	2018-10-30[18]	1.11.1	2019-01-11	No longer supported
1.12	2019-04-24[19]	1.12.2	2019-07-24	No longer supported
1.13	2019-10-30[20]	1.13.0	2019-10-30	No longer supported
1.14	2020-05-27[21]	1.14.2	2022-04-12	Fully supported, LTS

Legend: 🟥 Old version 🟨 Older version, still maintained 🟩 **Latest version** 🟧 Latest preview version

Release dates are extracted from Apache Subversion's `CHANGES` file,[22] which records all release history.

Features

- Commits as true atomic operations (interrupted commit operations in CVS would cause repository inconsistency or corruption).
- The system maintains versioning for directories and some specific file metadata (see Properties). Users can move or copy files and entire directory-trees very quickly, while retaining full revision history (as being implemented by a reference to the original object).
- Versioning of symbolic links.
- Native support for binary files, with space-efficient binary-diff storage.

- Apache HTTP Server as network server, WebDAV/Delta-V for protocol. There is also an independent server process called svnserve that uses a custom protocol over TCP/IP.
- Branching is implemented by a copy of a directory, thus it is a cheap operation, independent of file size.
- Natively client–server, layered library design.
- Client/server protocol sends diffs in both directions.
- Parsable output, including XML log output.
- Open source licensed – Apache License since the 1.7 release; prior versions use a derivative of the Apache Software License 1.1.
- Internationalized program messages.
- File locking for unmergeable files ("reserved checkouts").
- Path-based authorization.
- Language bindings for C#, PHP, Python, Perl, Ruby, and Java.
- Full MIME support – users can view or change the MIME type of each file, with the software knowing which MIME types can have their differences from previous versions shown.
- Merge tracking – merges between branches will be tracked, this allows automatic merging between branches without telling Subversion what does and does not need to be merged.
- Changelists to organize commits into commit groups.

Repository types

Subversion offers two types of repository storage.

Berkeley DB (deprecated)

The original development of Subversion used the Berkeley DB package. Subversion has some limitations with Berkeley DB usage when a program that accesses the database crashes or terminates forcibly. No data loss or corruption occurs, but the repository remains offline while Berkeley DB replays the journal and cleans up any outstanding locks. The safest way to use Subversion with a Berkeley DB repository involves a single server-process running as one user (instead of through a shared filesystem).[23] The Berkeley DB backend was deprecated in version 1.8.[24]

FSFS

In 2004, a new storage subsystem was developed and named FSFS. It works faster than the Berkeley DB backend on directories with a large number of files and takes less disk space, due to less logging.[23]

Beginning with Subversion 1.2, FSFS became the default data store for new repositories.

The etymology of "FSFS" is based on Subversion's use of the term "filesystem" for its repository storage system. FSFS stores its contents directly within the operating system's filesystem, rather than a structured system like Berkeley DB. Thus, it is a "[Subversion] FileSystem atop the FileSystem".

FSX

A new filesystem, called FSX, is under development to remove some limitations of FSFS. As of Version 1.9, it was not considered production-ready.[25]

Repository access

Access to Subversion repositories can take place by:

1. Local filesystem or network filesystem,[26] accessed by client directly. This mode uses the `file:///path` access scheme.
2. WebDAV/Delta-V (over http or https) using the `mod_dav_svn` module for Apache 2. This mode uses the `http://host/path` access scheme or `https://host/path` for secure connections using ssl.
3. Custom "svn" protocol (default port 3690), using plain text or over TCP/IP. This mode uses either the `svn://host/path` access scheme for unencrypted transport or `svn+ssh://host/path` scheme for tunneling over ssh.

All three means can access both FSFS and Berkeley DB repositories.

Any 1.x version of a client can work with any 1.x server. Newer clients and servers have additional features and performance capabilities, but have fallback support for older clients/servers.[27]

Layers

Internally, a Subversion system comprises several libraries arranged as layers. Each performs a specific task and allows developers to create their own tools at the desired level of complexity and specificity.

Fs

The lowest level; it implements the versioned filesystem which stores the user data.

Repos

Concerned with the repository built up around the filesystem. It has many helper functions and handles the various "hooks" that a repository may have, e.g., scripts that run when an action is performed. Together, Fs and Repos constitute the "filesystem interface".

mod_dav_svn

Provides WebDAV/Delta-V access through Apache 2.

Ra

Handles "repository access", both local and remote. From this point on, repositories are referred to using URLs, e.g.

- `file:///path/` for local access,
- `http://host/path/` or `https://host/path/` for WebDAV access, or
- `svn://host/path/` or `svn+ssh://host/path/` for the SVN protocol.

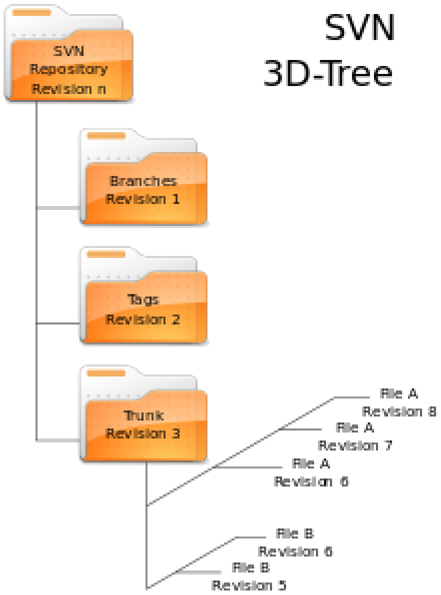
Client, Wc

The highest level. It abstracts repository access and provides common client tasks, such as authenticating users or comparing versions. Subversion clients use the Wc library to manage the local working copy.

Filesystem

One can view the Subversion filesystem as "two-dimensional".[28] Two coordinates are used to unambiguously address filesystem items:

- **Path** (regular path of Unix-like OS filesystem)
- **Revision**



Each revision in a Subversion filesystem has its own *root*, which is used to access contents at that revision. Files are stored as links to the most recent change; thus a Subversion repository is quite compact. The system consumes storage space proportional to the number of changes made, not to the number of revisions.

The Subversion filesystem uses transactions to keep changes atomic. A transaction operates on a specified revision of the filesystem, not necessarily the latest. The transaction has its own *root*, on which changes are made. It is then either committed and becomes the latest revision, or is aborted. The transaction is actually a long-lived filesystem object; a client does not need to commit or abort a transaction itself, rather it can also begin a transaction, exit, and then can re-open the transaction and continue using it. Potentially, multiple clients can access the same transaction and work together on an atomic change, though no existing clients expose this capability.

Properties

One important feature of the Subversion filesystem is properties: simple *name*=*value* pairs of text. Most properties occur on filesystem entries (i.e., files and directories). These are versioned just like other changes to the filesystem. The Subversion client reserves the 'svn:' prefix for built-in properties, but other names can be used to define custom properties.

`svn:executable`

Makes a file on Unix-hosted working copies executable, when supported by the filesystem.

`svn:mime-type`

Stores the Internet media type ("MIME type") of a file. Affects the handling of diffs and merging.

`svn:ignore`

A list of filename patterns to ignore in a directory. Similar to CVS's `.cvsignore` file.

`svn:keywords`

A list of *keywords* to substitute into a file when changes are made. The file itself must also reference the keywords as `$keyword$` or `$keyword:...$`. This is used to maintain certain information (e.g., author, date of last change, revision number) in a file without human intervention.

The keyword substitution mechanism originates from RCS and from CVS.[29]

`svn:eol-style`

Makes the client convert end-of-line characters in text files. Used when the working copy is needed with a specific EOL style. "native" is commonly used, so that EOLs match the user's OS EOL style. Repositories may require this property on all files to prevent inconsistent line endings, which can cause a problem in itself.

`svn:externals`

Allows parts of other repositories to be automatically checked out into a subdirectory.

`svn:needs-lock`

Specifies that a file is to be checked out with file permissions set to read-only. This is designed for use with the locking mechanism. The read-only permission reminds one to obtain a lock before modifying the file: obtaining a lock makes the file writable, and releasing the lock makes it read-only again. Locks are only enforced during a commit operation. Locks can be used without setting this property. However, that is not recommended, because it introduces the risk of someone modifying a locked file; they will only discover it has been locked when their commit fails.

`svn:special`

This property is not meant to be set or modified directly by users. As of 2010 it is only used for having symbolic links in the repository. When a symbolic link is added to the repository, a file containing the link target is created with this property set. When a Unix-like system checks out this file, the client converts it to a symbolic link.

`svn:mergeinfo`

Used to track merge data (revision numbers) in Subversion 1.5 (or later). This property is automatically maintained by the `merge` command, and it is not recommended to change its value manually.[30]

Subversion also uses properties on revisions themselves. Like the above properties on filesystem entries, the names are completely arbitrary, with the Subversion client using certain properties prefixed with 'svn:'. However, these properties are not versioned, and they can be changed later if allowed by a pre-revprop-change hook.[31]

`svn:date`

The date and time stamp of a revision.

`svn:author`

The name of the user that submitted the change(s).

`svn:log`

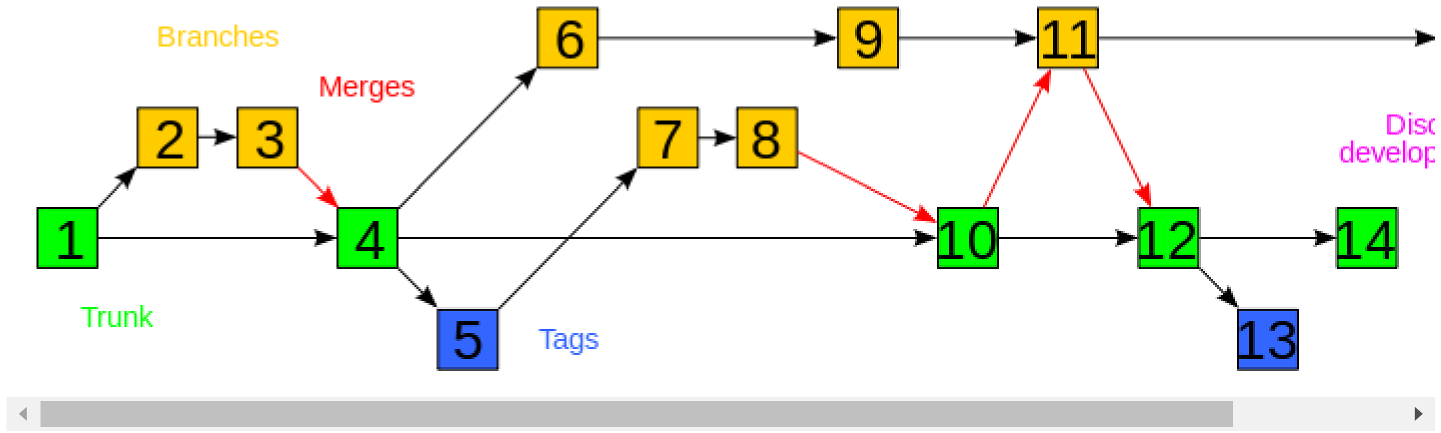
The user-supplied description of the change(s).

Branching and tagging

Subversion uses the inter-file branching model from Perforce[32] to implement branches and tagging. A branch is a separate line of development.[33] Tagging refers to labeling the repository at a certain point in time so that it can be easily found in the future. In Subversion, the only difference between branches and tags is how they are used.

A new branch or tag is set up by using the "`svn copy`" command, which should be used in place of the native operating system mechanism. The copied directory is linked to the original in the repository to preserve its history, and the copy takes very little extra space in the repository.

All the versions in each branch maintain the history of the file up to the point of the copy, plus any changes made since. One can "merge" changes back into the trunk or between branches.



Visualization of a simple Subversion project

Limitations and problems

A known problem in Subversion affects the implementation of the file and directory rename operation. As of 2014, Subversion implements the renaming of files and directories as a "copy" to the new name followed by a "delete" of the old name. Only the names change, all data relating to the edit history remains the same, and Subversion will still use the old name in older revisions of the "tree". However, Subversion may become confused when a move conflicts with edits made elsewhere,[34] both for regular commits and when merging branches.[35] The Subversion 1.5 release addressed some of these scenarios while others remained problematic.[36] The Subversion 1.8 release addressed some of these problems by making moves a first-class operation on the client, but it is still treated as copy+delete in the repository.[37]

As of 2013, Subversion lacks some repository-administration and management features. For instance, someone may wish to edit the repository to permanently remove all historical records of certain data. Subversion does not have built-in support to achieve this simply.[38]

Subversion stores additional copies of data on the local machine, which can become an issue with very large projects or files, or if developers work on multiple branches simultaneously. In versions prior to 1.7 these `.svn` directories on the client side could become corrupted by ill-advised user activity like global search/replace operations.[39] Starting with version 1.7 Subversion uses a single centralized `.svn` folder per working area.[40]

Subversion does not store the modification times of files. As such, a file checked out of a Subversion repository will have the 'current' date (instead of the modification time in the repository), and a file checked into the repository will have the date of the check-in (instead of the modification time of the file being checked in). This might not always be what is wanted.[41] To mitigate this, third-party tools exist that allow for preserving modification time and other filesystem meta-data.[42][43] However, giving checked out files a current date is important as well — this is how tools like make(1) will take notice of a changed file for rebuilding it.

Subversion uses a centralized revision control model. Ben Collins-Sussman, one of the designers of Subversion, believes a centralised model would help prevent "insecure programmers" from hiding their work from other team members.[44] Some users of version control systems see the centralised model as detrimental; famously, Linus Torvalds attacked Subversion's model and its developers.[45]

Subversion often does not deal well with the filename normalization performed by the HFS+ filesystem. This can cause problems when files with accented characters in their names are added to the repository on a non-HFS+ filesystem and the repository is then used with HFS+.[46]

Subversion tags and branches

Revision numbers are difficult to remember in any version-control system. For this reason, most systems offer symbolic *tags* as user-friendly references to them. Subversion does not have such a feature and what its documentation recommends to use instead is very different in nature. Instead of implementing tags as *references* to points in history, Subversion recommends making snapshot *copies* into a well-known subdirectory ("`tags/`") in the space of the repository tree. Only a few predefined *references* are available: `HEAD`, `BASE`, `PREV` and `COMMITTED`.

This history-to-space projection has multiple issues:

1. When a snapshot is taken, the system does not assign any special meaning to the name of the tag/snapshot. This is the difference between a *copy* and a *reference*. The revision is recorded and the snapshot can be accessed by URL. This makes some operations less convenient and others impossible. For instance, a naive `svn diff -r tag1:tag2 myfile` does not work; it is slightly more complicated than that to achieve, requiring the user to know and input URL/paths to the snapshots instead of just the names: `svn diff <URL-TO-TAG1>/myfile <URL-TO-TAG2>/myfile`. Other operations like for instance `svn log -r tag1:tag2 myfile` are just impossible.
2. When two (ideally independent) object types live in the repository tree, a "fight to the top" can ensue. In other words, it is often difficult to decide at which level to create the `tags/` subdirectory:

```
trunk/
     /componentfoo/
     /componentbar/
tags/
    /1.1/
        /componentfoo/
        /componentbar/
```

or

```
componentfoo/
            /trunk/
            /tags/
                 /1.1/
componentbar/
            /trunk/
            /tags/
                 /1.1/
```

3. Tags, by their conventional definition, are both read-only and light-weight, on the repository and client. Subversion copies are not read-only, and while they are light-weight on the repository, they are incredibly heavy-weight on the client.

To address such issues, posters on the Subversion mailing lists have suggested a new feature called "labels" or "aliases".[47] SVN labels would more closely resemble the "tags" of other systems such as CVS or Git. The fact that Subversion has global revision numbers opens the way to a very simple label → revision implementation. Yet as of 2013, no progress has been made and symbolic tags are not in the list of the most wanted features.[48]

Development and implementation

CollabNet has continued its involvement with Subversion, but the project runs as an independent open source community. In November 2009, the project was accepted into the Apache Incubator, aiming to become part of the Apache Software Foundation's efforts.[49] Since March 2010, the project is formally known as Apache Subversion, being a part of the Apache Top-Level Projects.[50]

In October 2009, WANdisco announced the hiring of core Subversion committers as the company moved to become a major corporate sponsor of the project. This included Hyrum Wright, president of the Subversion Corporation and release manager for the Subversion project since early 2008, who joined the company to lead its open source team.[51]

The Subversion open-source community does not provide binaries, but potential users can download binaries from volunteers.[52] While the Subversion project does not include an official graphical user interface (GUI) for use with Subversion, third parties have developed a number of different GUIs, along with a wide variety of additional ancillary software.

Work announced in 2009 included SubversionJ (a Java API) and implementation of the Obliterate command, similar to that provided by Perforce. Both of these enhancements were sponsored by WANdisco.[53]

The Subversion committers normally have at least one or two new features under active development at any one time. The 1.7 release of Subversion in October 2011 included a streamlined HTTP transport to improve performance and a rewritten working-copy library.[54]

In 2002, a design contest (http://svn.apache.org/repos/asf/subversion/svn-logos/README.html) was held to select the logo for Subversion. The original entries can be found here (http://svn.apache.org/repos/asf/subversion/svn-logos/entries.html) as well as the votes (http://svn.apache.org/repos/asf/subversion/svn-logos/VOTES) for each logo. The current logo received the most votes in the contest.

See also

foss ***Free and open-source software portal***

- List of version-control software
- Comparison of version-control software
- Comparison of Subversion clients
- List of software that uses Subversion
- TortoiseSVN

Notes

a. Apache-2.0 since 2009-07-07.

References

Footnotes

1. "Apache Subversion" (https://subversion.apache.org/news.html). Retrieved 12 February 2021.
2. "Subversion" (http://directory.fsf.org/wiki/Subversion#tab=Details). *directory.fsf.org*. Free Software Directory. 2013. Retrieved 11 September 2013.

3. "The Apache Software Foundation Announces 20th Anniversary of Apache® Subversion®" (https://www.globenewswire.com/news-release/2020/02/27/1991695/17401/en/The-Apache-Software-Foundation-Announces-20th-Anniversary-of-Apache-Subversion.html). *GlobeNewswire News Room* (Press release). 2020-02-27. Retrieved 2023-01-29.
4. Collins-Sussman, Ben; Brian W. Fitzpatrick; C. Michael Pilato (2011). "What is Subversion? > Subversion's History" (http://svnbook.red-bean.com/en/1.7/svn.intro.whatis.html#svn.intro.history). *Version Control with Subversion (for Subversion 1.7)*. Retrieved 15 March 2012.
5. Benjamin Zeiss (2004). "subversion 1.0 is released" (https://lwn.net/Articles/72498/). *Linux Weekly News*. Retrieved 30 March 2014.
6. Rubinstein, David (4 November 2009). "Subversion joins forces with Apache" (https://web.archive.org/web/20091111000653/http://www.sdtimes.com/SUBVERSION_JOINS_FORCES_WITH_APACHE/By_David_Rubinstein/About_APACHE_and_SUBVERSION/33886). SD Times. Archived from the original (http://www.sdtimes.com/link/33886) on 2009-11-11. Retrieved 15 March 2012.
7. "Subversion is now Apache Subversion" (https://web.archive.org/web/20110512171259/http://subversion.wandisco.com/component/content/article/1/43.html). 18 February 2010. Archived from the original (http://subversion.wandisco.com/component/content/article/1/43.html) on 12 May 2011. Retrieved 15 March 2012.
8. "Subversion 1.1 Release Notes" (https://subversion.apache.org/docs/release-notes/1.1.html). Retrieved 2015-09-21.
9. "Subversion 1.2 Release Notes" (https://subversion.apache.org/docs/release-notes/1.2.html). Retrieved 2015-09-21.
10. "Subversion 1.3 Release Notes" (https://subversion.apache.org/docs/release-notes/1.3.html). Retrieved 2015-09-21.
11. "Subversion 1.4 Release Notes" (https://subversion.apache.org/docs/release-notes/1.4.html). Retrieved 2015-09-21.
12. "Subversion 1.5 Release Notes" (https://subversion.apache.org/docs/release-notes/1.5.html). Retrieved 2015-09-21.
13. "Apache Subversion 1.6 Release Notes" (https://subversion.apache.org/docs/release-notes/1.6.html). Retrieved 2015-09-21.
14. "Apache Subversion 1.7 Release Notes" (https://subversion.apache.org/docs/release-notes/1.7.html). Retrieved 2015-09-21.
15. "Apache Subversion 1.8 Release Notes" (https://subversion.apache.org/docs/release-notes/1.8.html). Retrieved 2015-09-21.
16. "Apache Subversion 1.9 Release Notes" (https://subversion.apache.org/docs/release-notes/1.9.html). Retrieved 2015-09-21.
17. "Apache Subversion 1.10 Release Notes" (https://subversion.apache.org/docs/release-notes/1.10.html). Retrieved 2018-04-17.
18. "Apache Subversion 1.11 Release Notes" (https://subversion.apache.org/docs/release-notes/1.11.html). Retrieved 2018-10-31.
19. "Apache Subversion 1.12 Release Notes" (https://subversion.apache.org/docs/release-notes/1.12.html). Retrieved 2019-05-01.
20. "Apache Subversion 1.13 Release Notes" (https://subversion.apache.org/docs/release-notes/1.13.html).
21. "Apache Subversion 1.14 LTS Release Notes" (https://subversion.apache.org/docs/release-notes/1.14.html). *subversion.apache.org*. Retrieved 2020-06-02.
22. "Apache Subversion's CHANGES file" (https://svn.apache.org/repos/asf/subversion/trunk/CHANGES). Retrieved 2018-10-31.

23. Ben Collins-Sussman; Brian W. Fitzpatrick; C. Michael Pilato (2011). "Chapter 5: Strategies for Repository Deployment". *Version Control with Subversion: For Subversion 1.7* (http://svnbook.red-bean.com/en/1.7/svn.reposadmin.planning.html#svn.reposadmin.basics.backends). O'Reilly.
24. "Apache Subversion 1.8 Release Notes" (http://subversion.apache.org/docs/release-notes/1.8.html#bdb-deprecated). Apache Project. Retrieved 5 May 2020.
25. "Apache Subversion 1.9 Release Notes" (https://subversion.apache.org/docs/release-notes/1.9.html#fsx).
26. Berkeley DB relies on file locking and thus should not be used on (network) filesystems which do not implement them
27. SVN 1.5 release notes (http://subversion.apache.org/docs/release-notes/1.5.html)
28. Basic Merging (http://svnbook.red-bean.com/nightly/en/svn.branchmerge.basicmerging.html#svn.branchmerge.basicmerging.resurrect)
29. "rcs(1)" (https://man.openbsd.org/OpenBSD-current/man1/rcs.1). *OpenBSD manual pages*.
30. Subversion Properties (http://svnbook.red-bean.com/en/1.5/svn.ref.properties.html)
31. pre-revprop-change (http://svnbook.red-bean.com/en/1.7/svn.ref.reposhooks.pre-revprop-change.html)
32. Inter-File Branching: A Practical Method for Representing Variants (http://www.perforce.com/perforce/branch.html) Archived (https://web.archive.org/web/20070714160428/http://www.perforce.com/perforce/branch.html) 2007-07-14 at the Wayback Machine
33. Branching / Tagging — TortoiseSVN (http://tortoisesvn.net/docs/release/TortoiseSVN_en/tsvn-dug-branchtag.html)
34. "Implement true renames" (https://web.archive.org/web/20021201064344/http://subversion.tigris.org/issues/show_bug.cgi?id=898). Archived from the original (http://subversion.tigris.org/issues/show_bug.cgi?id=898) on 2002-12-01. Retrieved 2007-05-11.
35. Advanced Merging (http://svnbook.red-bean.com/en/1.7/svn.branchmerge.advanced.html#svn.branchmerge.advanced.moves)
36. Copy/move-related improvements in Subversion 1.5 (http://subversion.apache.org/docs/release-notes/1.5.html#copy-move-improvements)
37. Working copy records moves as first-class operation in Subversion 1.8 (http://subversion.apache.org/docs/release-notes/1.8.html#moves)
38. svn obliterate (http://subversion.tigris.org/issues/show_bug.cgi?id=516)
39. "Subclipse - Repair SVN Checksum" (https://stackoverflow.com/a/579442).
40. Working Copy Metadata Storage Improvements (client) (http://subversion.apache.org/docs/release-notes/1.7.html#wc-ng)
41. Issue 1256 (http://subversion.tigris.org/issues/show_bug.cgi?id=1256) at Tigris.org
42. FreezeAttrib (saves/restores file attributes using properties) (http://sourceforge.net/projects/freezeattrib/files/)
43. FSVS (Fast System VerSioning) (http://fsvs.tigris.org/)
44. Programmer Insecurity @ iBanjo (http://blog.red-bean.com/sussman/?p=96)
45. Google Tech Talk video (https://www.youtube.com/watch?v=4XpnKHJAok8) and its transcript (https://web.archive.org/web/20110725034129/https://git.wiki.kernel.org/index.php/LinusTalk200705Transcript)
46. subversion: Issue 2464 (http://subversion.tigris.org/issues/show_bug.cgi?id=2464)
47. For example: Weintraub, David (23 May 2005). "Subversion 'labels' vs. 'tags' " (https://svn.haxx.se/dev/archive-2005-05/1095.shtml). *SVN Dev* (Mailing list). Retrieved 5 May 2020.

48. "Apache Subversion Roadmap" (http://subversion.apache.org/roadmap.html). Apache Project. Retrieved 5 May 2020.
49. "Subversion Submitted to Become a Project at The Apache Software Foundation" (https://apache.org/foundation/press/pr_2009_11_04.html). November 4, 2009. Retrieved October 1, 2019.
50. "CollabNet Supports Subversion's Graduation to Apache Top-Level Project" (https://web.archive.org/web/20100311192650/http://www.open.collab.net/news/press/2010/apache.html). CollabNet. 1 March 2010. Archived from the original (http://www.open.collab.net/news/press/2010/apache.html) on 2010-03-11.
51. WANdisco (January 7, 2010). "WANdisco Names Hyrum Wright to Lead Subversion Open Source Efforts" (http://opensource.sys-con.com/node/1239202). *News release*. Open Source magazine. Archived (https://web.archive.org/web/20120118161238/http://opensource.sys-con.com/node/1239202) from the original on 2012-01-18. Retrieved October 29, 2011.
52. "Apache Subversion Binary Packages" (http://subversion.apache.org/packages.html). *Official project website*. Retrieved October 29, 2011.
53. WANdisco (October 28, 2009). "WANdisco Presents New Initiatives for the Subversion Open Source Project" (https://web.archive.org/web/20111118040455/http://www.cmcrossroads.com/index.php?Itemid=100152&catid=101%3Anews-and-announcements&id=13065%3Awandisco-presents-new-initiatives-for-the-subversion-open-source-project-&option=com_content&view=article). *News release*. CM Crossroads. Archived from the original (http://www.cmcrossroads.com/index.php?Itemid=100152&catid=101:news-and-announcements&id=13065:wandisco-presents-new-initiatives-for-the-subversion-open-source-project-&option=com_content&view=article) on 2011-11-18. Retrieved October 29, 2011.
54. "Apache Subversion Roadmap" (http://subversion.apache.org/roadmap.html). *Official project website*. Retrieved October 29, 2011.

Sources

- C. Michael Pilato, Ben Collins-Sussman, Brian W. Fitzpatrick; *Version Control with Subversion*; O'Reilly; ISBN 0-596-00448-6 (1st edition, paperback, 2004, full book online (http://svnbook.red-bean.com/), mirror (https://web.archive.org/web/20080107072118/http://www.mentalpointer.com/Subversion/svn-book.html))
- Garrett Rooney; *Practical Subversion*; Apress; ISBN 1-59059-290-5 (1st edition, paperback, 2005)
- Mike Mason; *Pragmatic Version Control Using Subversion*; Pragmatic Bookshelf; ISBN 0-9745140-6-3 (1st edition, paperback, 2005)
- William Nagel; *Subversion Version Control: Using the Subversion Version Control System in Development Projects*; Prentice Hall; ISBN 0-13-185518-2 (1st edition, paperback, 2005)

Further reading

- Dispelling Subversion FUD (http://www.red-bean.com/sussman/svn-anti-fud.html) by Ben Collins-Sussman (Subversion developer), link broken as of 2013-03-07 (Internet Archive.org Wayback Machine 2011-07-18 captured version (https://web.archive.org/web/20110718233416/http://www.red-bean.com/sussman/svn-anti-fud.html), "last updated" 2004-12-21)

External links

- Official website (https://subversion.apache.org/)

- Subversion (https://curlie.org/Computers/Software/Configuration_Management/Tools/Subversion/) at Curlie

Retrieved from "https://en.wikipedia.org/w/index.php?title=Apache_Subversion&oldid=1151083644"